The Report Committee for Gregory Alexander Williams
Certifies that this is the approved version of the following report:

River
A real-time location system providing indoor positional data acquisition for use in operational improvements within the healthcare environment.

APPROVED BY

SUPERVISING COMMITTEE:

Christine Julien, Supervisor

Constantine Caramanis

River
A real-time location system providing indoor positional data acquisition for use in operational improvements within the healthcare environment.

by

Gregory Alexander Williams

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2017

Dedicated to Gregory Hockenson, who never stopped believing in me.

Acknowledgments

I wish to thank Benjamin Celis and Jose Loyola, who helped develop the first iteration of River, Susan Pietrafeso, who gave us the idea, and Amanda McCarty, for her endless patience.

River
A real-time location system providing indoor positional data acquisition for use in operational improvements within the healthcare environment.

Gregory Alexander Williams, M.S.E.
The University of Texas at Austin, 2017

Supervisor: Christine Julien

Emergency Department (ED) over-crowding has enormous impact on both patient outcomes and reported satisfaction, costing hospitals thousands of dollars per year. Protocols for increasing efficiency have been proposed with varying justifications. Many consulting firms exist to help ERs identify problems and implement solutions, but no system exists specifically to benchmark all the relevant information in an automated fashion. This report describes such a system using Bluetooth Low Energy (BLE) technology implemented with beacon devices deployed in each area of interest within the ER and mobile applications that will be installed on personnel phones. This system is deployed in a physical setting, with results presented.

Table of Contents

Acknowledgments **v**

Abstract **vi**

List of Tables **ix**

List of Figures **x**

Chapter 1. Introduction **1**

1.1 Status of Emergency Care 1
1.2 RTLS Uses in Healthcare 2
1.3 Lean Practices for Improving Patient Flow 3
1.4 River System 5

Chapter 2. Literature Review **6**

2.1 IOM Report: Hospital-Based Emergency Care 6
2.2 Lean in the Emergency Department 8
2.3 Location Tracking in the Emergency Department 11

Chapter 3. Approach **13**

3.1 Physical Service 13
3.2 Detection Service 15
3.3 Storage Service 15
3.4 Creation Service 16
3.5 Service Interaction 17

Chapter 4. Implementation **18**
4.1 River iBeacon Scheme 18
4.2 River Mobile App 22
4.3 River Web 23
4.4 River API 23
4.5 Trial 24

Chapter 5. Results and Analysis **26**
5.1 First Experiment 26
5.2 Second Experiment 29
5.3 Third Experiment 31
5.4 Analysis 33

Chapter 6. Conclusion **34**

Appendices **36**

Appendix A. Mobile/Web Technology Stack **37**
A.1 River Mobile 37
A.2 River Web and API 38

Appendix B. Bluetooth 5.0 **40**
B.1 BLE 40

Appendix C. RTLS in Hospitals **42**
C.1 State of the Industry 42
C.2 Accomplishments 42
C.3 Hurdles 43

Bibliography **44**

Vita **48**

List of Tables

4.1 Floor plan to beacon major id mapping. 25

List of Figures

3.1 Service interaction. 17

4.1 Floor plan of trial location. 25

5.1 First experiment time sheet. 27
5.2 First experiment spaghetti diagram. 28
5.3 Second experiment time sheet. 30
5.4 Second experiment spaghetti diagram. 31
5.5 Third experiment time sheet. 32
5.6 Third experiment spaghetti diagram. 32

Chapter 1

Introduction

1.1 Status of Emergency Care

In 2006, a landmark report was released by the Institute of Medicine (IOM) that sought to examine "the full scope of emergency care, from 9-1-1 and medical dispatch to hospital-based emergency and trauma care[21]." The scale of the report is large and spans three published reports: *Hospital-Based Emergency Care: At the Breaking Point, Emergency Medical Services at the Crossroads, and Emergency Care for Children: Growing Pains.* Throughout the report recommendations are made on how to create the future of emergency care. One of the key recommendations in the first of the IOM reports was that "electronic dashboards and tracking systems" would be critical in facilitating the next generation of emergency department operations[21]. Most tracking systems in the healthcare environment are implemented as real-time location services (RTLS), and their applications are varied. All RTLS systems share the common goal of providing location information reliably at the level of granularity required for the particular use case.

Understanding and improving patient flow was another key recommendation in the report. Many studies have tried to determine the measurable clinical metrics that can help identify patient flow bottlenecks[25], and which operational techniques can be used to systematically remove them[13]. Ideas have been pulled from service and manufacturing industries and adapted to the healthcare environment. One promising framework is *Lean Sigma Six*, and there is a wealth of literature describing how *Lean Thinking* can be used to improve emergency departments[13, 14].

1.2 RTLS Uses in Healthcare

The use cases for RTLS can be categorized by what is being tracked. Patient, healthcare employee, and equipment tracking comprise the majority of applications[19]. Use cases can also be categorized based on the goal of the system; tracking equipment to help discover locations at the time of treatment, reducing time wasted searching, or tracking equipment to reduce procurements and facilitate maintenance. Tracking employees for inter-personnel communication, or examining the location data afterwards to determine bottlenecks in the processes of departments. The specific goal of the system will help define the requirements of the software and hardware components.

Many of these goals can be accomplished by one RTLS implementation, but large systems can often create more problems than they solve both in terms of employee compliance and performance. For example, an RTLS implementation attempted to integrate location tracking with a nurse call system and had nurses wear specialty badges. The nurses refused to wear them, rendering the system useless. Employee education and addressing privacy concerns is of paramount importance when implementing an RTLS system[19]. Another implementation faced performance concerns when a fundamental assumption about the system (ubiquitous WiFi) was not true, creating cost over-runs and preventing the system from bringing about the granular equipment tracking abilities that it promised[24].

RTLS systems are not a magic bullet for solving operational and inventory issues within the healthcare environment. The system needs to solve a concrete issue with non-, or minimally, invasive implementations. The technology is powerful, but without concrete goals the data generated by such systems can be useless.

1.3 Lean Practices for Improving Patient Flow

An overview of Lean Operations and Thinking, and the applications of lean methods in healthcare, is beyond the scope of this report, but most lean implementations in healthcare can be characterized by the following key principles:

- A focus on creating patient value
- Eliminating waste
- Promoting flow
- Continuously improving the process *with* the people

The focus of Lean Operations in promoting flow aligns very well with the IOM's recommendation to improve patient flow. Emergency departments looking to implement the report's recommendations could look to lean for concrete steps to take. One of the first concepts that Lean Operations introduce is "Standard Work[13]."

Standard Work is the act of defining the current work sequences in space and time[13] to determine if this aligns with staffing and patient demand. Part of Standard Work is the manual creation of descriptive artifacts that can help characterize work flows, mostly by passively watching personnel perform critical tasks and writing down observations. Many of these artifacts lend themselves to automatic creation, but two in particular are well suited toward RTLS systems: the Time Observation Worksheet, and the Spaghetti Diagram.

1.3.1 Time Observation Sheet

The Time Observation Sheet is a per-employee list of all the steps required for a specific process. Each step in the process is time-stamped over multiple iterations, and the results are recorded for later analysis.

1.3.2 Spaghetti Diagram

The Spaghetti Diagram maps the flow of personnel for a given process onto a picture of a working environment. This approximates the workflow of a particular activity.

Combining the Time Observation Sheet with the Spaghetti Diagram gives a complete picture of processes and helps identify bottlenecks and wasted time that does not add value.

1.4 River System

This report describes an RTLS system, named River, for use in emergency departments that can automate the creation of Time Observation Sheets and Spaghetti Diagrams for use in implementing lean practices. This report will review the current literature on implementing lean and RTLS systems within emergency departments. We will discuss the River system's attempt to combine RTLS systems and lean practices. How the River RTLS system will help enable lean practices, and River's implementation will be described. The report will present the results of a small-scale trial of the system in a representative environment. Then the report will detail the extensions of River and how it can help improve patient flow with future work.

Chapter 2

Literature Review

2.1 IOM Report: Hospital-Based Emergency Care

2.1.1 Relevant Challenges

The IOM report lists several over-arching challenges facing hospital-based emergency care centers, including crowding, ambulance diversion, inefficient use of resources, and lack of performance measurement and accountability.

Crowding occurs when patient volume backs up in emergency departments; patient flow is not high enough to allow new patients to be admitted in a timely manner. Crowding not only blocks access to emergency care, it also places stress on providers and patients alike[21].

Ambulance diversion is a challenge unique to the hospital-based emergency department. When emergency departments become saturated, inbound ambulances are re-routed. This used to be an unusual event, triggered by safety-value policies for extreme situations. However, federal studies report that over 500,000 ambulances were diverted in 2003. Nearly half of all hospitals in America report diversions occurring in 2004. The problem also introduces ethical concerns. For example, patients that are diverted may have worse outcomes than those who are not, leading one state (Massachusetts) to ban the practice altogether[16].

The IOM report cites inefficient use of inventory and personnel resources as a significant problem. The report referred to advancements in other industries that had produced operational tools and technologies to drive efficiency. Hospitals in particular have been slow to adopt new measures that could address bottlenecks in patient flow and improve efficiency[21].

A chronic lack of performance measurement and accountability means emergency departments are unable to assess how well they are serving their patients. The lack of standard definitions of performance and the technologies required to do site-by-site comparisons means that departments cannot provide any real understanding of the quality of care received.

2.1.2 Relevant Recommendations

The report outlines many operational improvement tools for dealing with bottlenecks to patient flow. It cites, in particular, *Lean Six Sigma's* use at Motorola and Toyota as inspirations for some of the recommended operations tools. Root-cause analysis and quality functional deployment are restatements of lean tools that require operational artifacts such as the Time Observation Sheet and Spaghetti Diagrams - artifacts that, through iterative analysis, help conflicting resource demands get resolved. Through root-cause analysis, these artifacts can highlight failure modes with the ED (perhaps one hallway is overused and resource locations can be moved to reduce time spent on tasks).

Regarding technological recommendations, the report calls for tracking technologies that provide the location of caregivers, resources, and possible patients. These systems can "empower administrators to understand how people move through the department[21]."

2.2 Lean in the Emergency Department

2.2.1 Focus on Creating Patient Value

Lean thought is interested in maximizing the value-delivery capabilities of any given system. Within emergency departments, the value proposition is providing patient care effectively and efficiently. In order to apply lean practices to the emergency department, patient value streams must be defined. These streams are groupings of patients that follow the same basic steps throughout their visit. In emergency settings, there are over-arching streams that deal with the general severity of the patient's condition and sub-streams that deal with the type of trauma or condition being treated. For instance, there could be high, mid, and low acuity patient streams with sub-streams for stroke symptoms, physical trauma, sepsis (life-threatening infection), or heart conditions. By defining these streams, one can map out the process for treating those patients. Within this well-defined map, each step is assigned one of three value-add codes: value-add, i.e. steps that a patient would classify as moving them towards wellness; non-value-add, i.e. steps that do not create patient-value (triage is a traditional example - patients come to see doctors and get diagnosed, not get interviewed by a nurse to see how long their treatment can wait); or business-value-add, i.e. steps that a patient would not consider value-add but must be done to keep the department working, such as billing and registration. Lean utilizes the mappings of all the patient streams and categorizes their steps into these value-add codes to help guide the reduction or elimination of non-value-add steps[13].

2.2.2 Eliminating Waste

Practitioners of lean know that not all non-value-add steps can be eliminated. But there are categorizations that can help guide process improvements to help eliminate as much as possible[13]. Any physical movement of people or goods is transportation waste. Time spent moving from patient to patient, or looking for supplies, is movement waste. Any supplies that are not being currently used to treat patients is inventory waste. Often in manufacturing or other industries, the reduction of unnecessary inventory is low-hanging fruit in terms of waste elimination[13]. In the emergency environment, where inventory may sit around for weeks before suddenly becoming critically needed, the remove of inventory is more complicated[14]. Waiting waste is when a patient is not being seen by a health-care professional. Anytime a provider does more than necessary to treat a patient is over-processing waste. Examples of over-processing include asking the same questions more than once or performing triage protocols that will be repeated by physicians. Over-production waste is any attempt to create value that is not necessary or realized. This includes ordering more tests that a patient's condition requires or prescribing unnecessary medication. When a test is not done correctly and must be repeated, or when a form needs to be filled out again due to clerical errors, defective waste is created. Whenever the front-line employees of a healthcare setting are not consulted for advice on how to improve processes, then there is a form of waste - the opportunity cost of not using available human resources can be very high[13].

2.2.3 Promoting Flow

Flow is the concerted effort to move patients from one value-add activity to the next with minimal waste. The ability to achieve high-quality flow in an emergency department has implications for the financial viability of the department but also carries clinical significance[13]. Departments with high clinical flow often have better outcomes for patients and lower re-admission rates[25].

2.2.4 Continuously Improving the Process *with* the People

Perhaps the most important aspect of lean is the use of continuous feedback, both from systems and from staff, to relentlessly pursue perfection in all value streams. Individual activities are also analyzed to perfect value streams. "Standard Work" aims to define a concrete time-limit for individual non-value-add activities that cannot be eliminated so that each stream can meet the expected demand. Artifacts, such as Time Observation Sheets and

Spaghetti diagrams are used to arrive are these concrete time-limits. These time-limits are always under scrutiny and can change as processes improve.

Often, front-line workers are empowered to change processes and acquire resources outside of the traditional chain of command. Using prescribed lean tools, such as value stream mapping, or standard work analysis, workers can often go from identifying a problem to addressing it in under a week[25].

2.3 Location Tracking in the Emergency Department

Many real-time location service (RTLS) systems used in hospital settings have been implemented. One popular implementation is Wi-Fi based RTLS[20]. This implementation has a low cost barrier and is relatively easy to implement physically. But accuracy is a chronic issue - Wi-Fi penetrates walls and can register false positives on tags within as large an area as 30 feet[20]. Other implementations use radio-frequency identification (RFID) tags which provide very granular location information but require extensive physical implementation, both for tagging people and resources of interest, but also in terms of readers that can identify the tags[8]. There are other Internet of Things style implementations, including ZigBee and Ultrasound capable tagging. There are trade-offs with these more esoteric technologies in terms of familiarity and available expertise.

Besides the different implementations, there are different system goals. Most RTLS installations are used to track assets - this application is important for cost-effectiveness, but the ideal RTLS system would be capable of tracking assets, personnel, and patients. Moreover, the ideal system would be able to contextualize this information automatically by answering *why* assets, personnel, or patients are in specific locations without manual data entry. The ideal RTLS implementation should "be scaled to meet a variety of tracking challenges across the locating/tracking/communications/workflow hierarchy[20]."

Chapter 3

Approach

In order to develop an effective RTLS system that can automatically create artifacts I developed River: a system for enabling lean practices in emergency departments. River consists of four separate services: 1) a physical service that provides a mapping of real-world locations to some defined set of logical identifiers; 2) a detection service that can be used to automatically determine which logical identifiers are currently relevant and feed this information to other services; 3) a storage and exposure service that stores logical location identifiers delivered by the detection service and exposes them to other services; 4) an artifact creation services that uses the identifier information to construct the artifacts of interest.

3.1 Physical Service

To expose physical locations programmatically, River uses iBeacon technology defined by Apple Inc[1], built on 4.2 BLE[1]. BLE beacons that follow the iBeacon specification are installed in fixed physical locations and periodically

[1]There are promises of beacon improvements in Bluetooth 5.0 that are covered in Appendix B

broadcast advertisement packets. The generic structure of a BLE packet is not covered here, but the BLE packets broadcast by an iBeacon compatible device contain three pieces of information: a Universally Unique Identifier (UUID), a “major” integer, and a “minor” integer. These three numbers uniquely identify each beacon. How these identifiers are defined and used to deliver functionality is unique to each deployment or system.

Apple places limitation of iOS devices that track iBeacons while the device is not active. In iOS devices are active (the screen is unlocked and responsive) applications can, in theory, receive periodic updates on the approximate distance of up to 4 *billion* iBeacons (called “ranging”). However, devices that are not active cannot range at all. What iOS does allows is for iBeacon *monitoring*, or the notification of entering the approximate area of iBeacons with a specific UUID, in the background. However, only 20 unique iBeacon UUIDs may be monitored at any given time. A detailed discussion of monitoring and its importance to River can be found in the Implementation chapter.

For River, the physical system is implemented with Estimote Location Beacons set to broadcast as iBeacons with unique UUIDs and a unique minor integer for each beacon. This allows for backgrounding-friendly (monitoring instead of ranging) schemes, which will be discussed further in implementation.

3.2 Detection Service

In order to determine personnel locations given the physical service, River uses smart-phone based beacon detection. This allows personnel to bring their own devices to the system and gain maximum value without large upfront costs and with minimal training. Most modern smart-phones have advanced Bluetooth chip-sets and software capabilities so this approach allows for the lowest barrier to adoption (no new devices, only an application). This also allows for the detection service to interface easily with the storage service. River's mobile application can detect beacons in the background and queue the location and time information so that it can be uploaded opportunistically. This does prevent the location data from being used in more immediate applications (for instance, signaling the status of a treatment room). However, the value of the data gathered does hold on longer time scales, and most uploads happen within minutes of the real event occurring in practice.

3.3 Storage Service

The storage service must be easily accessible by all other services. River uses an Internet application with an exposed REST API to provide access to all services for both reading and writing data.

3.3.1 Data Model

The data model starts with an organization. Each organization can have multiple sites. Each site defines their physical service mappings, and can contain multiple departments. Departments have personnel, which can record “stamps,” which contain relevant time and location data. This data model allows for many filtering options - by time across organization, site, department, or personnel, or by location across personnel. This provides for rich data representations and increases the number of artifacts that we can automatically generate to provide value to health organizations.

3.4 Creation Service

The creation services takes the recorded data across a defined time range and makes it available both visually and programmatically for analysis. The artifact focused on for this report is the Spaghetti Diagram needed for many lean operational improvements, but one can envision an entire dashboard of constantly updated data and artifact widgets for providing near-real-time insights into the operation of a particular health care department.

The creation service must talk with the storage service for more than artifact creation. New entity creation must also be supported through the creation service - when new organizations register for River, they will do so through the creation service. User credentials, security management, and data analysis will all be managed through the same Internet application. Future work could include adding creation service abilities in the detection service mobile application.

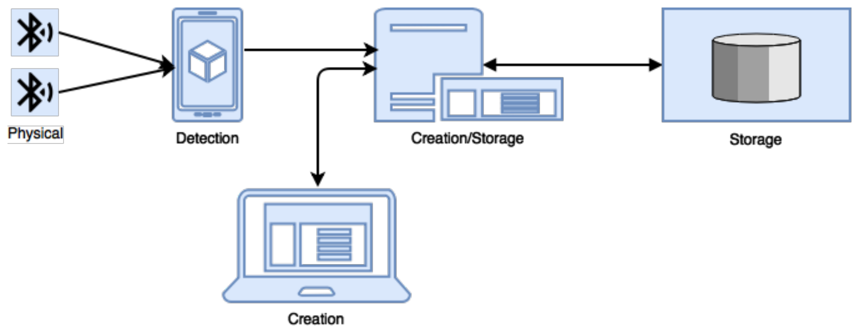

Figure 3.1: Service interaction.

3.5 Service Interaction

The services have a defined interaction scheme. The physical service is deployed into areas that need to provide location information. The detection service uses the storage service to determine the functional scheme used by the area the service is located in. Then the detection service uses the deployed physical service in tandem with the functional scheme to report relevant time and location stamps to the storage service. The creation service uses the storage service to access and filter the stamps based on the currently active user and creates artifacts based on that information as requested (Figure 3.1).

Chapter 4

Implementation

The implementation of River starts with the physical service, which uses a collection of Estimote Location Beacons set to broadcast as iBeacons. The other services are implemented with three separate projects, all using .NET Core 2.0.0. The website and API were created with ASP.NET Core, and the mobile application was built on Xamarin.

4.1 River iBeacon Scheme

4.1.1 iBeacon Monitoring and Ranging

In order to understand the implementation strategy of River's physical service, monitoring and ranging must be defined. They are the two most important technology services provided by the iBeacon protocol.

iBeacon monitoring refers to the idea of detecting any beacon that broadcasts a common UUID; this defines a "beacon region." Monitoring will alert the detection service whenever a beacon region is entered or exited. Entering a beacon region is only detected if the service begins outside of the region before monitoring was initiated. If monitoring is initiated inside of a beacon region, no entering event is recorded - the initial state must be requested explicitly and contextualized by the application. Exiting a beacon region is only detected after a time-delay upon physically exiting a region in order to prevent false positives.

iBeacon ranging refers to the idea of detecting all visible beacons and ranking them according to "approximate" distance. Ranging occurs on fixed intervals and returns a list of beacons. If there are no beacons visible when a ranging even occurs, an empty list is returned. Only beacons with a defined UUID of a registered region can be ranged.

In order to facilitate the testing and use of different scheme implementations, a per-platform beacon service is defined that accepts a beacon detection scheme.

4.1.2 Naive Scheme Implementation

The naive scheme implementation sets all the beacons in a particular location to have the same UUID, all the major IDs to be the same per department, and defines unique minor IDs for each beacon. Then the the detection service would only have to monitor for one region and could range for beacons to determine which department and room is detected. Whenever the closest beacon that is ranged corresponds to a different location than the currently marked location, the detection service would create a new stamp, upload that stamp to the storage service, and mark this new location as the current location.

There are a few problems with this approach. River implements the detection service with a mobile application, and on one of the most popular mobile application platforms, Apple iOS. iOS has very strict rules for what can and cannot be done in the background of an application. Of particular interest to River's physical service implementation is the limitation on beacon ranging and HTTP communication in the background[1].

In order to work around these limitations, a different approach must be taken. The over-all strategy is to use iBeacon monitoring services to notify the application of entrance to a room by tying unique iBeacon information to physical locations. Specifically, the notification upon entering an iBeacon region will wake up the application (regardless of whether the app has been closed, moved to the background, or the phone is locked) and allow a timed (non-deterministic, but near 10 seconds[18]) bit of computation to execute. iOS applications are only allowed to monitor for a maximum of 20 beacon regions, and so a strategy for shuffling the regions being monitored during our brief period of computation must be implemented.

4.1.3 5 Colors Scheme Implementation

A 5 color scheme implementation will rely on the fact that the beacons are arranged in a planar map, so for the purposes of the physical scheme it is assumed that only one floor will be tracked at any time. The extension to multiple floors is easy (attached planes where moving between floors represent a border), but requires more coding in the scheme strategies. According to the

[1]See Appendix A

Four Color Theorem, “given any separation of a plane into contiguous regions, producing a figure called a map, no more than four colors are required to color the regions of the map so that no two adjacent regions have the same color.[23]” This means that each beacon region could be assigned one of 4 unique UUIDs and be assured that you would get a monitoring entrance event to a different UUID before the current UUID was seen again. The proof of this is fairly complex and the problem generated many false contradictions[9], but there are quadratic algorithms to generate mappings[23].

There are, however, linear algorithms that exist for generating the mapping if we are allowed to use 5 colors[23]. Speed does not matter for the current implementation of River, but future versions could include the ability to add new tracked locations on the fly, necessitating a re-mapping quickly and easily. Since the scheme only uses 5 of the available 20 monitoring regions, the possibility for designing a more robust scheme exists - layering two mappings on top of each other so that if one beacon fails, the other scheme would still correctly locate the staff within the corresponding room.

In researching implementations, a short-coming of the 5 color scheme was discovered. Because beacon regions have the possibility of overlapping, and monitoring events are not exclusive, determining the precise region that was being currently monitored (and therefore representative of the correct location) was not guaranteed[17]. Another approach yields better real-world results for correct location determination in small scale studies.

4.1.4 Nearest Neighbors Scheme Implementation

The nearest neighbors scheme is a bit less elegant then the 5 colors scheme. It involves giving every beacon in a tracked location a unique UUID and uses the 10 seconds of computation time to change the currently monitored regions to the 20 nearest neighbors. Since the monitoring regions is destroyed and rebuilt each time an enter event is fired, the application uses more battery and needs more configuration - schemes must be stored locally, and in the case of iOS, not encrypted, so that the list of available regions is ready to update in the background.

In small scale applications this scheme is more accurate, but issues with scaling are easy to spot. For example, this scheme does not allow for higher density beacon schemes for increased location accuracy - each beacon already has a unique UUID so layering two implementations would involve tracking two separate neighbor-maps and deciding at runtime what location was indicated. The metric for deciding between the two maps is not obvious.

4.2 River Mobile App

The detection service is implemented with a mobile app on iOS[2]. It has a platform specific "Beacon Service" that takes a scheme as a strategy and reports "hits" as location updates, writing the relevant information to disk to be upload at the earliest available opportunity. In the current implementation,

[2]For full technology stack, see Appendix A

the mobile app is relatively barebones, given a list of the other personnel in the department ranked by distance. But the app could eventually be a nexus for the creation of new floor-plan schemes, secure inter-personnel communication, and feedback mechanisms to further enable lean tools. Since the devices on which the application is built are advancing quickly, and since the Bluetooth 5.0 specification includes sweeping changes to the amount of information a BLE device can emit and the communication that can be enable[3], future work for the app could include merging with the physical service and providing more accurate timing and location information.

4.3 River Web

The creation service is implemented with Microsoft's ASP.net Core and hosted on Azure Web Application service. The data is pulled from the River API and JavaScript is used to generate the Spaghetti Diagram and Time Observation Worksheets.

4.4 River API

The storage service is a REST API hosted on Microsoft's Azure Web Application service. This allows for client-oriented navigation of the API's available endpoints and the backing-store's relationship data. The data is stored in an Azure SQL Database instance. The API allows for quick access

[3] See Appendix B

to data and simplifies the reading and writing of complex relationship data. The API also defines the data model used by all other services, using the json:api specification[2] to define the API's response format.

A full treatment of the software technology stack for all River services is given in Appendix A.

4.5 Trial

A small trial was run in an office space setting. Six Estimote Beacons, set to broadcast iBeacon packets using the Nearest Neighbors scheme, were placed in six locations. The River front end and mobile app had the beacon scheme hard-coded to facilitate the experiment. The space had floor maps available for use in overlaying the Spaghetti Diagram (Figure 4.1). Each experiment represented a unique "task".

The three experiments are a defined sequence of locations and time delays, and have the resulting Time Observation Sheet and Spaghetti Diagrams compared with the expected values. A mobile device had the River mobile app installed and was walked between 6 locations (Table 4.1).

Figure 4.1: Floor plan of trial location.

Table 4.1: Floor plan to beacon major id mapping.

Map Location	Beacon Major	Symbol
4S.D03	3505	
CONF. 4S13	4500	
4S.E20L	19053	
4S.E02	19350	
Unmarked room South of CONF. 4S13	36288	
4S.E11	36646	

Chapter 5

Results and Analysis

5.1 First Experiment

5.1.1 Sequence and Timing

The first experiment was two circular trips between beacon-identified locations, pausing for 2 minutes at each location in the sequence:

1. 4S.E20L
2. 4S.E02
3. 4S.D03
4. CONF. 4S13
5. Unmarked room South of CONF. 4S13
6. 4S.E11

A perfectly accurate RTLS system would be expected to record this sequence twice.

5.1.2 Results

The expected sequence of locations was never recorded, making the generated Time Observation Sheet (figure 5.1) more of a guideline and less of an operations tool. The Spaghetti Digram is more useful, as the overall picture

Personnel	Location	Time (UTC)
FirstExperiment	19053	11/22/2017 4:27:57 AM
FirstExperiment	19350	11/22/2017 4:27:57 AM
FirstExperiment	19053	11/22/2017 4:27:58 AM
FirstExperiment	19350	11/22/2017 4:27:58 AM
FirstExperiment	19053	11/22/2017 4:27:58 AM
FirstExperiment	19350	11/22/2017 4:27:58 AM
FirstExperiment	19053	11/22/2017 4:27:58 AM
FirstExperiment	19350	11/22/2017 4:27:59 AM
FirstExperiment	19053	11/22/2017 4:27:59 AM
FirstExperiment	19350	11/22/2017 4:27:59 AM
FirstExperiment	19053	11/22/2017 4:27:59 AM
FirstExperiment	3505	11/22/2017 4:32:50 AM
FirstExperiment	19350	11/22/2017 4:35:28 AM
FirstExperiment	4500	11/22/2017 4:35:40 AM
FirstExperiment	3505	11/22/2017 4:38:00 AM
FirstExperiment	4500	11/22/2017 4:39:20 AM
FirstExperiment	3505	11/22/2017 4:41:20 AM
FirstExperiment	19053	11/22/2017 4:44:12 AM
FirstExperiment	19350	11/22/2017 4:46:46 AM
FirstExperiment	36646	11/22/2017 4:46:56 AM
FirstExperiment	19350	11/22/2017 4:47:21 AM
FirstExperiment	36288	11/22/2017 4:49:33 AM
FirstExperiment	3505	11/22/2017 4:50:15 AM

Figure 5.1: First experiment time sheet.

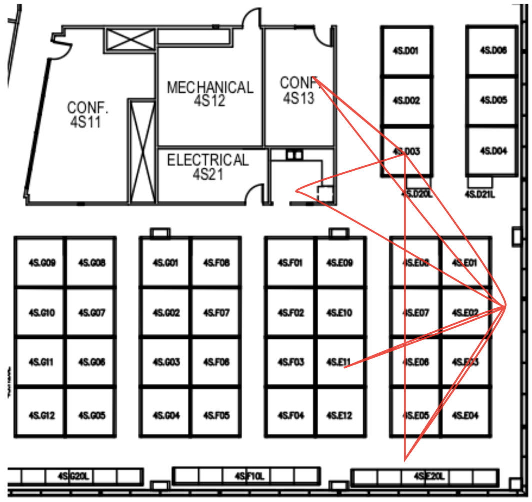

Figure 5.2: First experiment spaghetti diagram.

(the flow of the activity) can be inferred from the diagram (Figure 5.2) if one disregards “impossible” paths. However, it is clear that the walls separating the rooms in the floor plan are not successfully blocking the BLE packets from traversing through and so false-positives are largely present. This problem is particularly prevalent for two locations: 4S.E20L and 4S.E02. These locations were being registered as the device was moving through connecting hallways. Moreover, the time between location changes (2 minutes) allowed for more packet arrivals that, given the nearest neighbors scheme, could register a false positive in location change. This is most obvious in the first set of time stamps, seen as a repeated back and forth oscillation between 4S.E20L and 4S.E02. The back-and-forth within one minute shows that these were not true location changes, but flaws in the beacon detection scheme.

5.2 Second Experiment

5.2.1 Sequence and Timing

The second experiment was the same sequence of steps in two circular trips, but only pausing for 1 minute at each location. If the major contributor to the false positives was the beacon packets bleeding through walls, this experiment should result in fewer false positives (less time to see a false packet). However, if the scheme itself was to blame (by creating false positives as it switched to monitoring for neighbor UUIDs), the same amount of false positives would be expected.

5.2.2 Results

This is much closer to correct than the first experiment, and the first sequence is only off by two mis-placed stamps (Figure 5.3). However, the oscillation between neighbor locations is still present (this time between Unmarked room South of CONF. 4S13 and 4S.E11, at the end of the experiment). So the miscues probably have a shared responsibility - the location is being falsely identified both by stray packets and by the nearest neighbors scheme itself. This can be seen by looking at the Spaghetti Diagram (Figure 5.4) - 4S.E02 still has impossible paths attributed to it, so stray packets are playing a part. But because the oscillating between neighbors is present, the scheme also is not as accurate as required.

Personnel	Location	Time (UTC)
SecondExperiment	19053	11/22/2017 4:53:57 AM
SecondExperiment	3505	11/22/2017 4:54:42 AM
SecondExperiment	19350	11/22/2017 4:54:58 AM
SecondExperiment	4500	11/22/2017 4:55:14 AM
SecondExperiment	36288	11/22/2017 4:55:44 AM
SecondExperiment	3505	11/22/2017 4:57:40 AM
SecondExperiment	19350	11/22/2017 5:00:37 AM
SecondExperiment	36646	11/22/2017 5:00:49 AM
SecondExperiment	19350	11/22/2017 5:01:53 AM
SecondExperiment	19053	11/22/2017 5:04:43 AM
SecondExperiment	3505	11/22/2017 5:05:32 AM
SecondExperiment	19053	11/22/2017 5:06:00 AM
SecondExperiment	19350	11/22/2017 5:06:09 AM
SecondExperiment	3505	11/22/2017 5:06:10 AM
SecondExperiment	36288	11/22/2017 5:06:15 AM
SecondExperiment	3505	11/22/2017 5:06:39 AM
SecondExperiment	36288	11/22/2017 5:07:07 AM
SecondExperiment	19350	11/22/2017 5:07:18 AM
SecondExperiment	36646	11/22/2017 5:07:23 AM
SecondExperiment	36288	11/22/2017 5:07:58 AM

Figure 5.3: Second experiment time sheet.

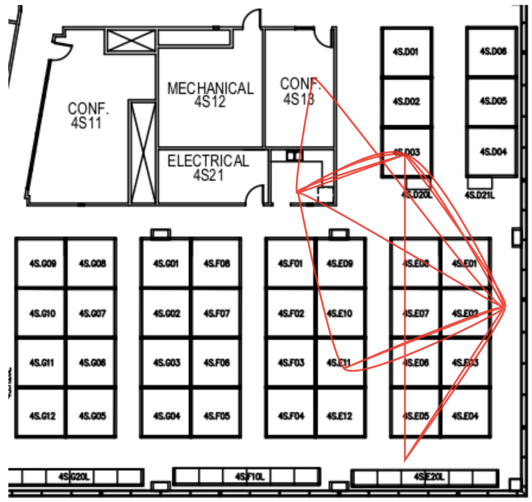

Figure 5.4: Second experiment spaghetti diagram.

5.3 Third Experiment

5.3.1 Sequence and Timing

The third experiment was the same two cycles in the same sequence, this time with no pause at each location. Hopefully with minimal time for stray beacon packet detection, the location scheme might work correctly.

5.3.2 Results

This was the closest to accurate among all the experiments (Figure 5.5). But there was one glaring issue - 4S.E02 did not register at all during the experiment (Figure 5.6). Again, the current implementation can be useful for generating "approximate" artifacts, like the Spaghetti Diagram, but not so useful for application requiring precision in the indoor location.

Personnel	Location	Time (UTC)
ThirdExperiment	19053	11/22/2017 5:09:09 AM
ThirdExperiment	3505	11/22/2017 5:09:33 AM
ThirdExperiment	4500	11/22/2017 5:10:07 AM
ThirdExperiment	36288	11/22/2017 5:10:17 AM
ThirdExperiment	19053	11/22/2017 5:10:32 AM
ThirdExperiment	3505	11/22/2017 5:12:01 AM
ThirdExperiment	36288	11/22/2017 5:12:12 AM
ThirdExperiment	3505	11/22/2017 5:12:40 AM
ThirdExperiment	36646	11/22/2017 5:12:46 AM

Figure 5.5: Third experiment time sheet.

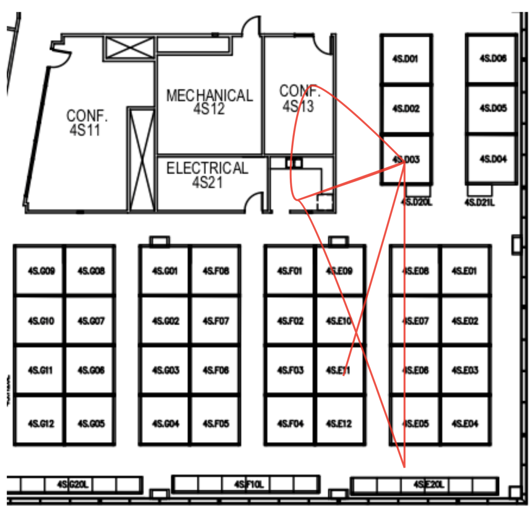

Figure 5.6: Third experiment spaghetti diagram.

5.4 Analysis

The beacons all set to the same settings and placed within cubicles did not provide the level of accuracy that a Time Observation Sheet requires. However, the overall level of accuracy was enough for Spaghetti Diagrams, which are defined as "approximations" anyway.

More interesting is the difficulty in defining a detection scheme that creates accurate location estimates. Nearest neighbors did an ok job when the beacons were seen quickly and when the number of neighbors was low, but seems to have issues simply because every time a device sees a beacon it must change which beacons it is monitoring for. Without a more robust location scheme, River cannot be an industry-ready system.

Chapter 6

Conclusion

The need for improved operations at hospital-based emergency centers to improve patient flow and adequately serve patient populations is clear from available reports. RTLS systems can be a powerful tool to help in many areas of concern. But the need for a clear goal for the RTLS system to reduce adoption friction, and the engineering needed to produce the accuracy necessary are difficult barriers to entry. There are many technologies that enable RTLS systems[26]. Their use is continuing to grow in real-world healthcare settings[15]. By looking for unique value propositions within the context of RTLS technologies and improvements to healthcare operations, opportunities for new business cases and technology applications can be found.

This paper presented River, a system for generating Lean operations artifacts automatically as an RTLS system. This is a clear goal that drives towards operational improvements. However, the artifacts created were not perfectly accurate and can only be described as useful for "approximate" artifacts. River is an attempt to bridge the operations world and RTLS technologies directly. There are many technological reasons why River's current implementation is not enough, including beacon monitoring scheme limitations and iOS backgrounding limitations. River can approximately determine location within a real setting but is not deterministic enough to be used for anything other than approximate needs. Lean Spaghetti Diagrams are approximate enough; time observation sheets are not. The nearest neighbors scheme is not robust enough in its current implementation.

Future work for researchers looking at bridging new operations techniques and RTLS systems could include looking for outside location service technology providers, such as the Estimote Indoor SDK, or putting more robust engineering effort into refining the location schemes, to improve accuracy and reliability. The ability to rapidly deploy location services to new areas without available maps could be explored. The ability to rapidly introduce new features, such as equipment location or room-status updates, to the River system is another avenue of investigation.

Appendices

Appendix A

Mobile/Web Technology Stack

A.1 River Mobile

A.1.1 iOS

The River Mobile App is currently operational for iOS only, so iPhones are the main mobile device. iPads could also be used.

A.1.1.1 Core Location

Core Location provides services for determining a device's geographic location, altitude, orientation, or position relative to a nearby iBeacon. The framework uses all available onboard hardware, including Wi-Fi, GPS, Bluetooth, magnetometer, barometer, and cellular hardware to gather data[1]. The Core Location APIs most relevant to River is the iBeacon services. Applications can use the Core Location classes to listen for and react to the presence of iBeacons, which is the basis for most indoor location detection on iOS devices.

A.1.2 Xamarin

Xamarin is a framework by Microsoft that allows developers to write applications in C# and have the resulting applications run natively on iOS devices, as well as Android and Windows Phone devices[6]. The framework provides wrappers for all of the native platform APIs as well as mechanisms for wrapping other libraries that are provided as iOS Objective-C or Swift libraries, as well and .jar libraries meant to run on Android. This allows for a large amount of code to be shared across all major mobile platforms.

Xamarin is based on the Mono runtime, an implementation of the .NET framework designed to run on Linux, MacOS, iOS, and Android. River is written in C# using the Xamarin framework.

A.1.3 DI/IoC with Autofac

Inversion of Control (IoC) is a software pattern where the classes that have dependencies do not instantiate the concrete implementations; instead, dependencies are given to them[3]. This allows developers to replace or update the dependency implementation without affecting the original class. Dependency Injection is a mechanism for giving classes the dependencies that they require. Autofac is a popular .NET IoC container to facilitate Dependency Injection. The River Mobile App uses Autofac for Dependency Injection.

A.2 River Web and API

A.2.1 ASP.NET Core

ASP.NET Core is a open-source framework for constructing web-based applications. It is the foundation of both the River Web and River API implementations. ASP.NET Core provides many benefits as a modern web framework including built-in Dependency Injection, and support for most popular front-end frameworks including AngularJS and React. Both the River Web and River API applications are built and served with ASP.NET Core.

A.2.2 Azure

Azure is Microsoft's cloud provider framework, with advanced tooling for web-based applications, cloud storage, and continuous integration and deployment. Both River Web and API are deployed on Azure.

A.2.3 d3.js

d3.js is a JavaScript library for manipulating DOM elements using data. It is a minimal library, not a monolithic data-representation framework, but it aims to "solve the crux of the problem: efficient manipulation of documents based on data.[12]" d3.js is used to draw the spaghetti diagrams through River Web based on location data generated by beacons and River Mobile.

A.2.4 JSON API .Net Core

JSON API .Net Core is a framework for building json:api compliant web servers. It is based on ASP.NET Core and allows for quickly creating RESTful APIs based on model data. The River API is based on JSON API .Net Core and serves json:api compliant documents.

Appendix B

Bluetooth 5.0

The new Bluetooth 5.0 Specification includes a wide array of new features and enhancements, including extended range and increased data transfer speeds, however, the improvements that matter to River are the sweeping changes in how BLE can operate.

B.1 BLE

The BLE physical layer has been redefined so that compliant radios will have the ability to send 2 megabits per second, double that of the Bluetooth 4.2 specification. If BLE radios want to send the same amount of information, say in advertising packets, they can now send the packets twice as fast. The additional bandwidth/transmission speed comes from a reworking of the data modulation, not any increase in power usage, so these benefits come without additional power usage costs. Compliant radios will also have the ability to send packets on a new second BLE physical layer called "LE Coded" which sends the same old standard of 1 megabits per second with a lower bit encoding and higher power usage to increase the range of packets up to four times previous ranges. But by far the most important change for River to the BLE specification is the changes to advertising itself.

Older Bluetooth LE radios can only advertise in the “advertisement channels,” three dedicated channels each 2MHz wide. There are 40 defined channels for Bluetooth LE communication - the remaining 37 channels are reserved for data connections. New Bluetooth LE radios will have the ability to “chain” advertisement packets to packets in the data channels, effectively allowing BLE radios to stream data to any listening devices. The new packets in the data channels (called “secondary advert channels” in the spec) will have a new header that is defined so that listening devices that use older specs will simply discard these new packets, but now the context that a BLE device can broadcast is much more robust, and listeners can effectively “tune-in” to entire streams of data[5].

New beacons can take advantage of this by not only streaming more context, but also by being listeners of BLE streams, allowing the BLE devices to respond to their environment, not just reflect it. Powerful, lightweight, meshed networks of beacons can reflect and respond to context near-continuously with BLE 5.0 advertisement chaining.

Appendix C

RTLS in Hospitals

C.1 State of the Industry

The current adoption of RTLS systems in hospitals is concentrated around RFID tags for equipment and employee id badges, currently comprising 43% of all RTLS installations[4]. The global industry as a whole is expected to see a compound annual growth rate of over 50% between 2017 and 2021[4]. The value of automatic asset tracking is starting to take shape as early adopters practices change to take advantage of process improvements[4]. The expected size of the global market will reach $8.09 billion by the end of the growth period[4].

C.2 Accomplishments

Many hospitals are starting to reap the benefits of RTSL even after initial setbacks[15]. Asset tracking is lowering inventory costs and starting to increase equipment availability[15].

C.3 Hurdles

Most of the hurdles for RTLS revolve around two omni-present factors in such systems - accuracy and ill-defined problems. Accuracy for RTLS systems is a big deal, if the system generates data that is incorrect, not only has the system not offloaded work from personnel, it has added work: the data must be entered manually and any incorrect data must be corrected. Defining the problem being solved is another big issue. Often times, when RTLS systems were extended past equipment tracking to include personnel and patients, there was an active sabotage of the system[15]. Privacy invasion without a clear benefit to a defined and prevalent problem will continue to make adoption of personnel and patient tracking difficult.

Bibliography

[1] Core location. *Core Location — Apple Developer Documentation.* https://developer.apple.com/documentation/corelocation.

[2] JSON API latest specification (v1.0). *JSON API*, May 2015. http://jsonapi.org/format/.

[3] Getting started with autofac. *Getting Started - Autofac 4.0 documentation*, Feb 2017. http://autofac.readthedocs.io/en/latest/getting-started/index.html.

[4] Global rtls market in healthcare sector to grow at a cagr of over 502021, says technavio. *Investment Weekly News*, page 412, 2017.

[5] What's new with bluetooth 5.0? *Laird Technologies Wireless Connectivity Blog*, Feb 2017. http://www.summitdata.com/blog/whats-new-with-bluetooth-5-0/.

[6] Xamarin. *Introduction to Mobile Development - Xamarin*, Mar 2017.

[7] Apple Inc. *Location and Maps Programming Guide*, Mar 2016.

[8] Beth Bandi. Rtls - to be or not to be. Technical report, EDI, Ltd., 2011.

[9] Frank R. Bernhart. A digest of the four color theorem. *Journal of Graph Theory*, 1(3):207–225, 1977.

[10] Bluetooth SIG. *Bluetooth Core Specifications*, v4.2 edition, 2014.

[11] Bluetooth SIG. *Bluetooth Core Specifications*, v5.0 edition, 2016.

[12] Mike Bostock. Data-driven documents. *D3.js*, 2017. https://d3js.org/.

[13] Jody Crane and Chuck Noon. *The definitive guide to emergency department operational improvement: employing lean principles with current ED best practices to create the "no wait" department*. Productivity Press, New York, 2011.

[14] H. de Koning, J. P. S. Verver, J. van den Heuvel, S. Bisgaard, and R. J. M. M. Does. Lean six sigma in healthcare. *Journal for Healthcare Quality*, 28(2):4–11, 2006.

[15] JA Fisher and T. Monahan. Evaluation of real-time location systems in their hospital contexts. *INTERNATIONAL JOURNAL OF MEDICAL INFORMATICS*, 81(10):705–712, 2012.

[16] Joel M. Geiderman, Catherine A. Marco, John C. Moskop, James Adams, and Arthur R. Derse. Ethics of ambulance diversion. *The American journal of emergency medicine*, 33(6):822, 2015;2014;.

[17] Devney Hamilton. CoreLocation in the Wild: Experiments in monitoring more than 20 iBeacon regions. *Atomic Spin*, April 2016. https://spin.atomicobject.com/2016/04/18/monitoring-over-20-ibeacon-regions/.

[18] Apple Inc. App programming guide for ios, Mar 2017.

[19] Maged N. Kamel Boulos and Geoff Berry. Real-time locating systems (rtls) in healthcare: a condensed primer. *International journal of health geographics*, 11(1):25–25, 2012.

[20] Rick Krohn. The optimal RTLS solution for hospitals. *J Healthc Inf Manag*, 23(2):4–5, 2009.

[21] Committee on the Future of Emergency Care in the United States Health System and Institute of Medicine (U.S.). Committee on the Future of Emergency Care in the United States Health System. *Hospital-based emergency care: at the breaking point*. National Academies Press, Washington, D.C, 2006–2007.

[22] John Reilly. The emergency department: Rising wait times, costs, and possible remedies. Master's thesis, The University of Texas School of Public Health, 2010.

[23] Neil Robertson, Daniel Sanders, Paul Seymour, and Robin Thomas. Efficiently four-coloring planar graphs. In *Proceedings of the twenty-eighth annual ACM symposium on theory of computing*, pages 571–575. ACM, 1996.

[24] Jeremy Schwartz. Va's foray into internet of things faced catastrophic failure. *Austin American-Statesman*, Jun 2017.

[25] Amith L. Shetty, Caleb Teh, Matthew Vukasovic, Shannon Joyce, Milan R. Vaghasiya, and Roberto Forero. Impact of emergency department discharge stream short stay unit performance and hospital bed occupancy rates on access and patient flow measures: A single site study. *Emergency Medicine Australasia*, 29(4):407–414, 2017.

[26] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A survey of indoor localization systems and technologies. *CoRR*, abs/1709.01015, 2017.

Vita

Gregory Alexander Williams was born in Syracuse, New York on 28 December 1986, the son of Sander Williams and Susan Hockenson. He received a Bachelor of Science degree in Physics from The University of Texas at Austin. While in school he worked as a software technician for the Quest Learning and Assessment program. After graduating, he went to work for National Instruments as a Software Engineer. While employed with National Instruments, he applied to the Cockrell School of Engineering's Engineering Executive Education program. He was accepted and started graduate studies in August, 2015.

Address: grw300@gmail.com

This report was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.